EXPRESS MAIL LABEL NO:

EV 047 534 011 US

This application is submitted in the name of the following inventors:

Inventor	Citizenship	Residence City and State
Bechtolsheim, Andreas V.	Germany	Stanford, California
Cheriton, David R.		

The assignee is Cisco Technology, Inc., a California corporation having an office at 170 West Tasman Drive, San Jose CA 95134.

Title of the Invention

Access Control List Processing in Hardware

Background of the Invention

1. Field of the Invention

This invention relates to access control list processing.

2. Related Art

In a computer network for transmitting information, messages can be restricted from being transmitted from selected source devices to selected destination devices. In known computer networks, this form of restriction is known as "access control" and is performed by routers, which route messages (in the form of individual packets of information) from source devices to destination devices. One known technique for access control is for each router to perform access control by reference to one or more ACLs (access control lists); the ACL describes which selected source devices are permitted (and which denied) to send packets to which selected destination devices.

In a known standard for ACL format, each ACL includes a plurality of access control specifiers, each of which selects a range of sender and destination IP address prefix or subnet, and port, and provides that packet transmission from that selected set of senders to that selected set of destinations is either specifically permitted or specifically denied. ACLs are associated with input interfaces and independently with output interfaces for each router. In known routers such as those manufactured by Cisco Systems, Inc., of San Jose, California, the router is provided with an ACL using an ACL command language, interpreted by operating system software for the router, such as the IOS operating system.

One problem in the known art is that processing of packets to enforce access control according to the ACL is processor-intensive and can therefore be relatively slow, particularly in comparison with desired rates of speed for routing packets. This problem is exacerbated when access control is enforced for packets using software in the router, because software processing of the ACL can be quite slow relative to hardware processing of the packet for routing.

One known solution is to reduce the number of packets for which access control requires actual access to the ACL. In a technique known as "netflow switching," packets are identified as belonging to selected "flows," and each packet in a flow is expected to have identical routing and access control characteristics. Therefore, access control only requires reference to the ACL for the first packet in a flow; subsequent packets in the same flow can have access control enforced identically to the first packet, by reference to a routing result cached by the router and used for the entire flow.

Netflow switching is further described in detail in the following patent applications:

- U.S. Application Serial No. 08/581,134, titled "Method For Traffic Management, Traffic Prioritization, Access Control, and Packet Forwarding in a Datagram Computer Network", filed December 29, 1995, in the name of inventors David R.

Cheriton and Andreas V. Bechtolsheim, assigned to Cisco Technology, Inc., attorney docket number CIS-019;

- U.S. Application Serial No. 08/655,429, titled "Network Flow Switching and Flow Data Export", filed May 28, 1996, in the name of inventors Darren Kerr and Barry Bruins, and assigned to Cisco Technology, Inc., attorney docket number CIS-016; and

- U.S. Application Serial No. 08/771,438, titled "Network Flow Switching and Flow Data Export", filed December 20, 1996, in the name of inventors Darren Kerr and Barry Bruins, assigned to Cisco Technology, Inc., attorney docket number CIS-017.

These patent applications are collectively referred to herein as the "Netflow Switching Disclosures". Each of these applications is hereby incorporated by reference as if fully set forth herein.

While netflow switching achieves the goal of improving the speed of enforcing access control by the router, it still has the drawback that comparing at least some incoming packets against the ACL must be performed using software. Thus, the relative slowness required by software processing of the ACL is not completely avoided.

Summary of the Invention

The invention provides a method and system for hardware processing of ACLs and thus hardware enforcement of access control. A sequence of access control specifiers from an ACL are recorded in a CAM, and information from the packet header is used to attempt to match selected source and destination IP addresses or subnets, ports, and protocols, against all the ACL specifiers at once. Successful matches are input to a priority selector, which selects the match with the highest priority (that is, the match that is first in the sequence of access control specifiers). The specified result of the selected match is used to permit or deny access for the packet without need for software processing, preferably at a rate comparable to wirespeed.

In a preferred embodiment, the CAM includes an ordered sequence of entries, each of which has an array of ternary elements for matching on logical "0", logical "1", or on any value, and each of which generates a match signal. The ACL entered for recording in the CAM can be optimized to reduce the number of separate entries in the CAM, such as by combining entries which are each special cases of a more general access control specifier.

A router including the CAM can also include preprocessing circuits for certain range comparisons which have been found both to be particularly common and to be otherwise inefficiently represented by the ternary nature of the CAM. For example,

comparisons of the port number against known special cases, such as "greater than 1023" and "within the range 6000 to 6500", can be treated by circuitry for performing range comparisons or by reference to one or more auxiliary CAMs.

The invention can also be used to augment or override routing decisions otherwise made by the router, so as to implement QOS (quality of service), and other administrative policies, using the CAM.

Brief Description of the Drawings

Figure 1 shows a block diagram of a system for access control list processing.

Figure 2 shows a block diagram of an access control element.

Figure 3 shows a flow diagram of a method for access control list processing in hardware.

Detailed Description of the Preferred Embodiment

In the following description, a preferred embodiment of the invention is described with regard to preferred process steps and data structures. Those skilled in the art

would recognize after perusal of this application that embodiments of the invention can be implemented using circuits adapted to particular process steps and data structures described herein, and that implementation of the process steps and data structures described herein would not require undue experimentation or further invention.

System Elements

Figure 1 shows a block diagram of a system for access control list processing.

A system 100 includes a set of packet input interfaces 101, a routing element 110, an access control element 120, and a set of packet output interfaces 102. The system 100 receives packets 130 at the input interfaces 101; each packet 130 indicates a source device 131, from which it was sent, and a destination device 132, to which it is intended to go. The routing element 110 processes each packet 130 to select one or more of the output interfaces 102 to which the packet 130 should be forwarded. The access control element 120 determines if the packet 130 has permission to be forwarded from its source device 131 to its destination device 132. Each packet 130 that has permission to be forwarded is output to its selected output interfaces 102.

In a first set of alternative embodiments, the system 100 may include a plurality of access control elements 120 operating in parallel in place of the single access control element 120.

In a second set of alternative embodiments, the system 100 may include one or more access control elements 120 coupled to the input interfaces 101 and operating to determine if packets 130 have permission to be forwarded from their source devices 131 at all. The access control element 120 is shown coupled to the routing element 110 to perform access control after a routing decision has been made. However, the access control element 120 is still capable of denying access to packets 130 responsive to whether they have permission to be forwarded from their source devices 131 at all.

In a third set of alternative embodiments, the system 100 may include one or more access control elements 120 coupled to individual input interfaces 101 and operating to make access control determinations for packets 130 arriving at particular input interfaces 101. Similarly, the system 100 may include one or more access control elements 120 coupled to individual output interfaces 102 and operating to make access control determinations for packets 130 forwarded to particular output interfaces 102.

Access Control Element

Figure 2 shows a block diagram of an access control element.

In a preferred embodiment, the access control element 120 operates on a set of selected elements of a packet header 133 for each packet 130. The system 100 collects the selected elements into a packet label 200.

In a preferred embodiment using netflow switching, the packet label 200 used for access control at the input interfaces 101 includes a source device 131, the destination device 132, a port identifier for a port at the source device 131, a port identifier for a port at the destination device 132, and a protocol type. In alternative embodiments, the packet label 200 may be any collection of information derived from the packet 130 (preferably from the packet header 133) used for access control.

The concept of preprocessing the packet label has wide applicability, including determining other routing information in response to data in the packet header. For example, in addition to or instead of comparing data in the packet header against known special cases, such as "greater than 1023" and "within the range 6000 to 6500," preprocessing can include performing logical or arithmetic operations on data in the packet header. Preprocessing can also include data lookup, or substituting new data, in response to data in the packet header.

The access control element 120 includes an input port 201 coupled to the packet label 200, an access control memory 210, a priority encoder 220, and an output port 202 coupled to the priority encoder 220.

When the access control element 120 is disposed for controlling access for packets responsive to their input interfaces 101, the packet label 200 includes an identifier for the input interface 101. When the access control element 120 is disposed for controlling access for packets responsive to their output interfaces 102, the packet label 200 includes an identifier for the output interface 102.

The access control memory 210 includes a CAM (content-addressable memory) having a sequence of access control specifiers 211. Each access control specifier 211 includes a label match mask 212 and a label match pattern 213. For each access control specifier 211, each bit of the label match mask 212 determines whether or not a corresponding bit of the packet label 200 is tested. If so, the corresponding bit of the label match pattern 213 is compared for equality with the corresponding bit of the packet label 200. If all compared bits are equal, the access control specifier 211 matches the packet label 200. Bits that are not compared have no effect on whether the access control specifier 211 is considered to match the packet label 200 or not.

The priority encoder 220 is coupled to all of the access control specifiers 211, and receives an indicator from each one whether or not that access control specifier

211 matched the packet label 200. The priority encoder 220 selects the single access control specifier 211 with the highest priority (in a preferred embodiment, the one with the lowest address in the access control memory 210) and provides an indicator of that single access control specifier 211 to the output port 202.

The indicator provided to the output port 202 specifies whether or not the packet 130 has permission to be forwarded from its specified source device 131 to its specified destination device 132. In a preferred embodiment, the indicator specifies one of three possibilities: (a) the packet 130 is forwarded to its calculated output interface and on to its specified destination device 132; (b) the packet 130 is dropped; or (c) the packet 130 is forwarded to a "higher-level" processor for further treatment. When a packet 130 is dropped it is effectively denied access from its specified source device 131 to its specified destination device 132.

The higher-level processor includes a general-purpose processor, program and data memory, and mass storage, executing operating system and application software for software (rather than hardware) examination of the packet 130. The packet 130 is compared, possibly to the access control specifiers 211 and possibly to other administrative policies or restrictions, by the higher-level processor. The higher-level processor specifies whether the packet 130, after processing by the higher-level processor, is forwarded to a selected output interface or is dropped.

Access Control Lists

A Cisco access control list includes a sequence of access control entries, which are mapped to a set of access control specifiers 211. Each access control entry has a structure according to the following syntax:

access-list *access-list-number* [**dynamic** *dynamic-name* [**timeout** *minutes*]] {**deny|permit**} *protocol source source-wildcard* [*operator port* [*port*]] *destination destination-wildcard* [*operator port* [*port*]] [**established**] [**precedence** *precedence*] [**tos** *tos*] [**log**]

This syntax, its meaning, and access control entries in general, are further described in documentation for Cisco IOS software, available from Cisco Systems, Inc., in San Jose, California, and hereby incorporated by reference as if fully set forth herein.

Access control entries can specify that particular actions are permitted, denied, or that they will be recorded in a log. Access control entries are interpreted sequentially. Thus, an earlier more specific access control entry can prohibit particular actions (such as receiving messages from a particular sending device), while a later more general access control entry can permit the same actions for other devices (such as other sending devices in the same network).

When an access control list is translated for entry into the access control memory, it is optimized to reduce the number of separate entries that are used. Thus, an access control list with N separate access control entries is translated into a set of access control specifiers 211 that can be smaller or larger than N, depending on the effect of optimization.

A first optimization detects separate access control entries that each refer to a special case of a more general access control specifier 211, such as in one of the following cases:

- A first access control entry provides a selected permission for a selected source device 131 2S, and a second access control entry provides the same permission for a selected source device 131 2S+1. The first and second access control entries can be translated into a single more general access control specifier 211 with an unmatched bit in the 2^0 position.

- A set of access control entries each provides the same selected permission for a range of selected source devices 131 S through T, and the range S through T can be represented as a smaller number of bit strings with unmatched bits.

- A set of access control entries provides a selected permission for a comparison of source device 131 addresses with a test value V.

A second optimization detects range comparisons that have been found to be particularly common. For example, it is common to compare the source or destination port number for being greater than 1023, or for being within the range 6000 to 6500. To compare the source or destination port number for being greater than 1023 with matched and unmatched bits would use about six entries for each such comparison (to test each one of the six high-order bits of the port number for being logical "1").

In a preferred embodiment, a comparison circuit 230 compares the source port number and the destination port number with these known ranges and provides a set of comparison bits 231 indicating whether or not the source port number and the destination port number are within each specified range. The comparison circuit 230 includes a finite state machine 232 (or other element) for storing lower and upper bounds for each specified range. The comparison bits 231 are coupled to the input port 201 of the access control element 120 for treatment as matchable input bits supplemental to the header of the packet 130.

In various embodiments, the invention can be used to augment or override routing decisions otherwise made by the router, using the access control element 120. In addition to specifying that the packet 130 is to be dropped or forwarded to the higher-level processor, the access control element 120 can alter the output interface, which was selected by the routing element 110, to another selected output interface. The invention

can thus be used to implement QOS (quality of service) policies and other administrative policies.

Method of Operation

Figure 3 shows a flow diagram of a method for access control list processing in hardware.

A method 300 includes a set of flow points to be noted, and steps to be executed, cooperatively by the elements of the system 100.

At a flow point 310, a packet is received at one of the packet input interfaces 101.

At a step 321, the routing element 110 receives an input packet 130.

At a step 322, the routing element 110 identifies the header for the packet 130.

At a step 323, the routing element 110 selects portions of the header for use as the packet label 200 for access control. In a preferred embodiment, the packet label 200 used for access control at the input interfaces 101 includes the source device 131, the

destination device 132, the port identifier at the source device 131, the port identifier at the destination device 132, and a protocol type.

At a step 324, the routing element 110 couples the packet label 200 and an input interface specifier to the input access control element 120.

At a step 325, the routing element 110 determines a selected output interface for the packet 130.

At a step 326, preferably performed in parallel with the step 325, the input access control element 120 determines the input permission for the packet 130, that is, whether the routing element 110 permits forwarding the packet 130 from the source device 131 for the packet 130.

The step 326 includes matching the packet label 200 against the access control memory 210 for the input access control element 120, determining all of the successful matches, coupling the successful matches to the priority encoder 220 for the input access control element 120, determining the highest-priority match, and providing an output result from the input access control element 120.

If at the step 326, the input access control element 120 determines that the higher-level processor should process the packet 130, the higher-level processor proc-

esses the packet 130. A result from the higher-level processor is substituted for the result from the input access control element 120.

If at the step 326, the input access control element 120 (or the higher-level processor) determines that the packet 130 should be dropped, the packet 130 is dropped, and the routing element 110 takes no further action with regard to the packet 130.

At a step 327, the routing element 110 couples the packet label 200 and the output interface specifier to the output access control element 120.

At a step 328, the output access control element 120 determines the output permission for the packet 130, that is, whether the routing element 110 permits forwarding the packet 130 to the destination device 132 for the packet 130.

The step 326 includes the following actions:

- matching the packet label 200 against the access control memory 210 for the output access control element 120;
- determining all of the successful matches;

- coupling the successful matches to the priority encoder 220 for the output access control element 120;

- determining the highest-priority match; and

- providing an output result from the output access control element 120.

If at the step 328, the output access control element 120 determines that the higher-level processor should process the packet 130, the higher-level processor processes the packet 130. A result from the higher-level processor is substituted for the result from the output access control element 120.

If at the step 328, the output access control element 120 (or the higher-level processor) determines that the packet 130 should be dropped, the packet 130 is dropped, and the routing element 110 takes no further action with regard to the packet 130.

At a flow point 330, the packet is ready for transmission to one of the packet output interfaces 102.

Alternative Embodiments

Although preferred embodiments are disclosed herein, many variations are possible which remain within the concept, scope, and spirit of the invention, and these variations would become clear to those skilled in the art after perusal of this application.